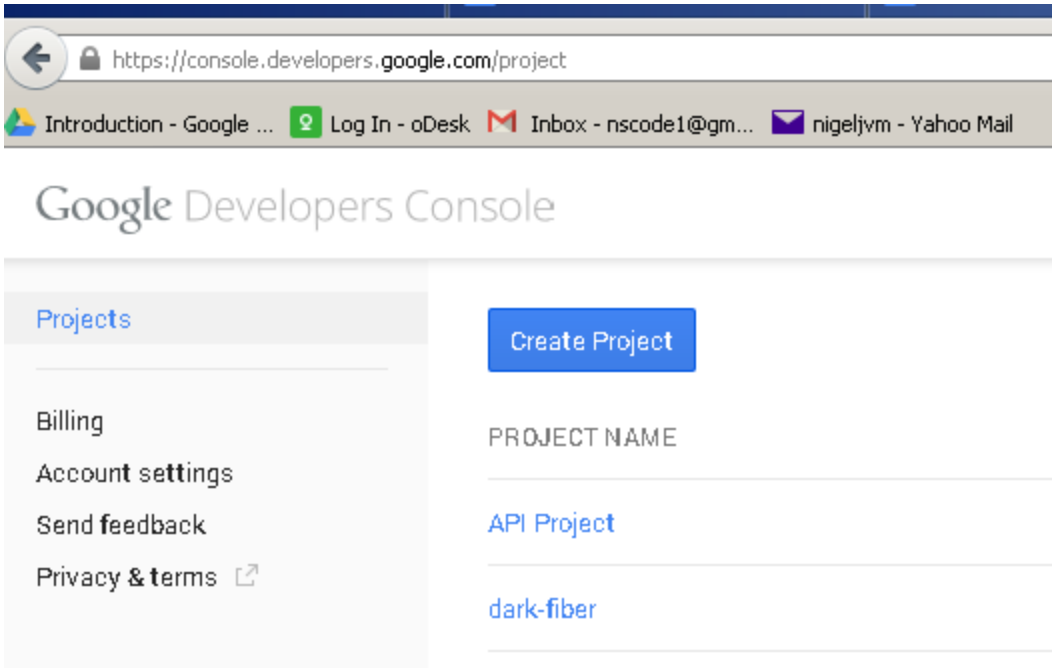
(IaaS) (PaaS) Google Cloud Compute

Google Cloud Platform Services

Step1

Sign in to developers console and create an application
https://console.developers.google.com/project



Once you click on “Create” button, It will ask for SMS verification to prevent the abuse.
Go through the process of SMS verification.

Click on the application and it will bring up the page that shows the possible application types

Click on the “Compute Engine” and it will ask for Enable Billing. You need to enable the billing

Step 2

Choose an instance
These are the possible machines

An instance that lives for 11.25 minutes will be charged for 12 minutes of usage.

US | Europe | Asia

Standard machine types

Configuration	Virtual Cores	Memory (GB[1])	GCEUs what is this?	Local Disk (GB)	Price (USD) / Hour
n1-standard-1	1	3.75	2.75	0	$0.07
n1-standard-2	2	7.50	5.50	0	$0.14
n1-standard-4	4	15	11	0	$0.28
n1-standard-8	8	30	22	0	$0.56
n1-standard-16	16	60	44	0	$1.12

Configuration	Virtual Cores	Memory (GB)	GCEUs what is this?	Local Disk (GB)	Price (USD) / Hour
f1-micro	1	0.60	Shared CPU, not guaranteed	0	$0.013
g1-small	1	1.70	1.38	0	$0.035

Step 3

Select your application with billing enabled and then select to the right “Spin up a Cloud Compute Engine VM”

Project Dashboard

Boost your app with a Google API

Use Google APIs with your app to harness the power of Google's services and technologies

Enable an API

Launch click-to-deploy software

Deploy popular open stacks on Google Compute Engine just by filling out a simple form

Deploy Cassandra on Compute Engine
Deploy MongoDB on Compute Engine
Deploy the Ruby Stack on Compute Engine
Deploy the MEAN Development Stack on Compute Engine
Deploy RabbitMQ on Compute Engine
Deploy GitLab Community Edition (CE) on Compute Engine

Spin up a Compute Engine VM

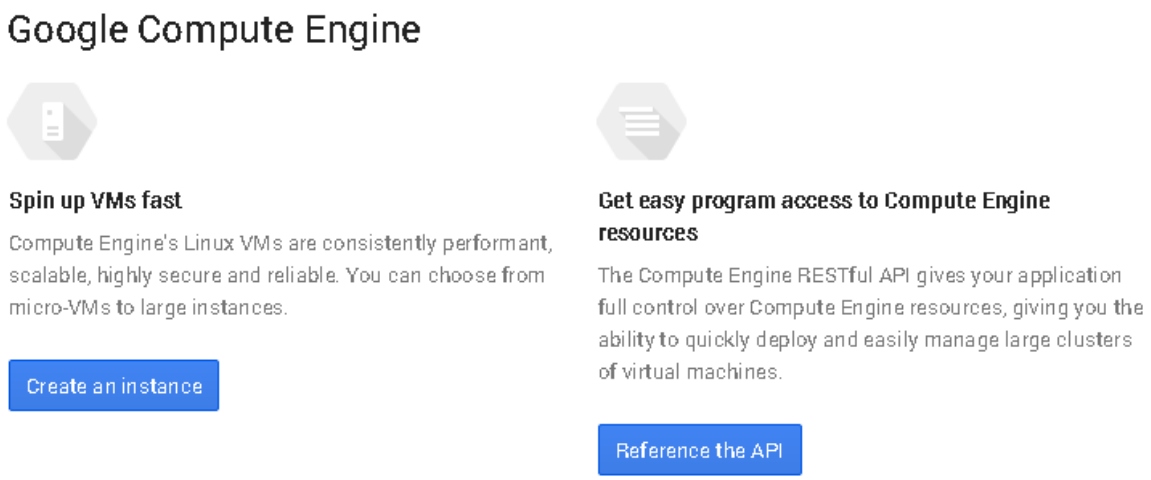
Get started with Google Compute Engine by c first VM instance

Create instance

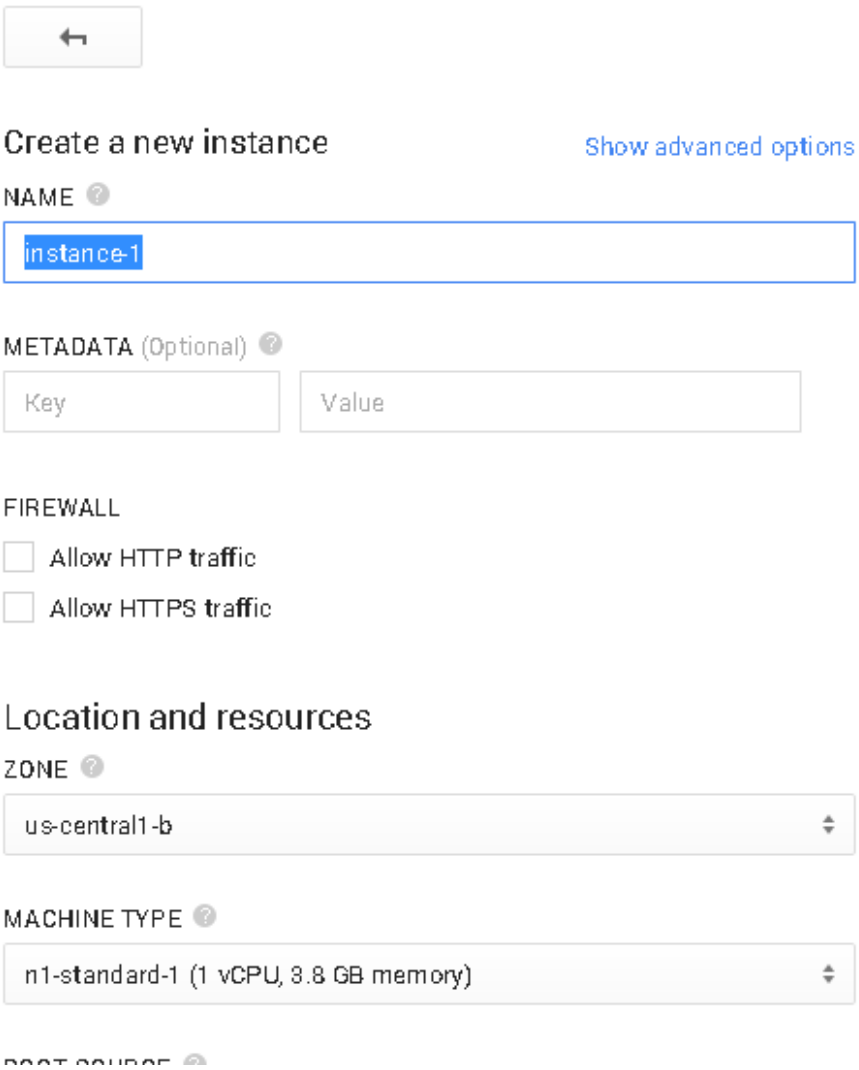
you may get this message

Google Compute Engine is not ready for use yet in the project. It may take several minutes if Google Compute Engine has just been enabled. Compute Engine in the project.

wait for a while and then refresh, repeat this until you see the screen below



select create an instance you should then see the screen below



Name : Name of the Instance, this name will be used with in the Google VM Instances for communication as a DNS name
Tags : tags to your instance as EC2
Metadata : Every instance stores its metadata on the metadata server.
Regions & Zones : as EC2

Machine Type : see 'These are the possible machines' chart above

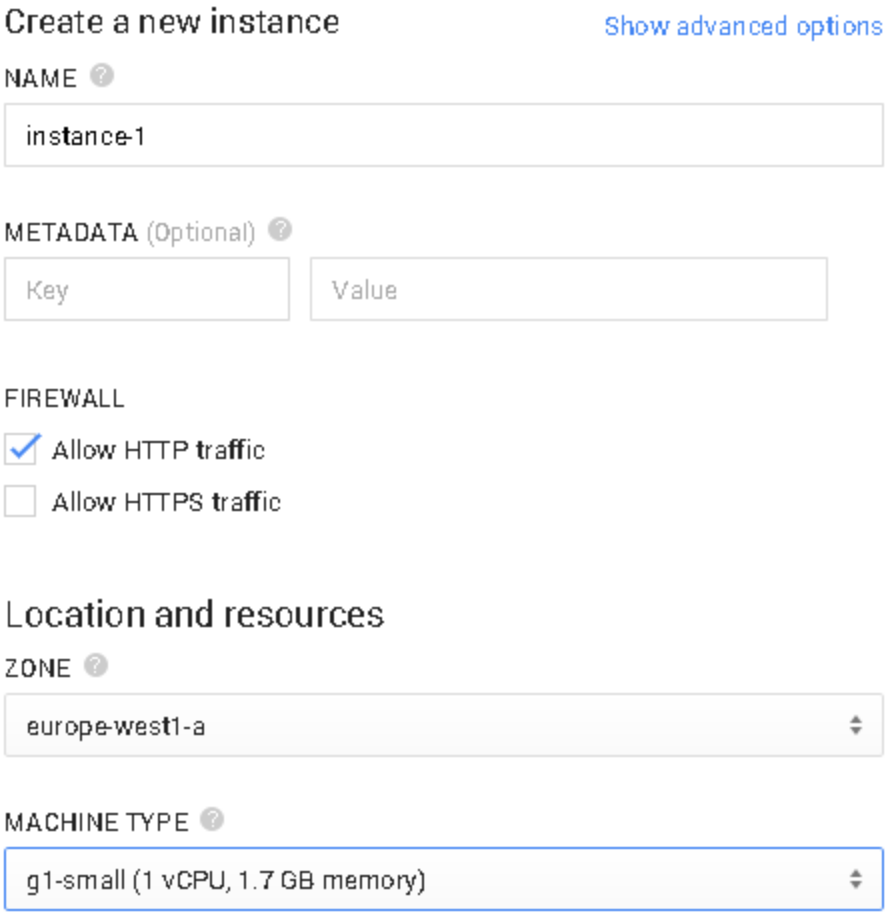
Boot Source : It determines the disk used to boot the instance. Keep the default

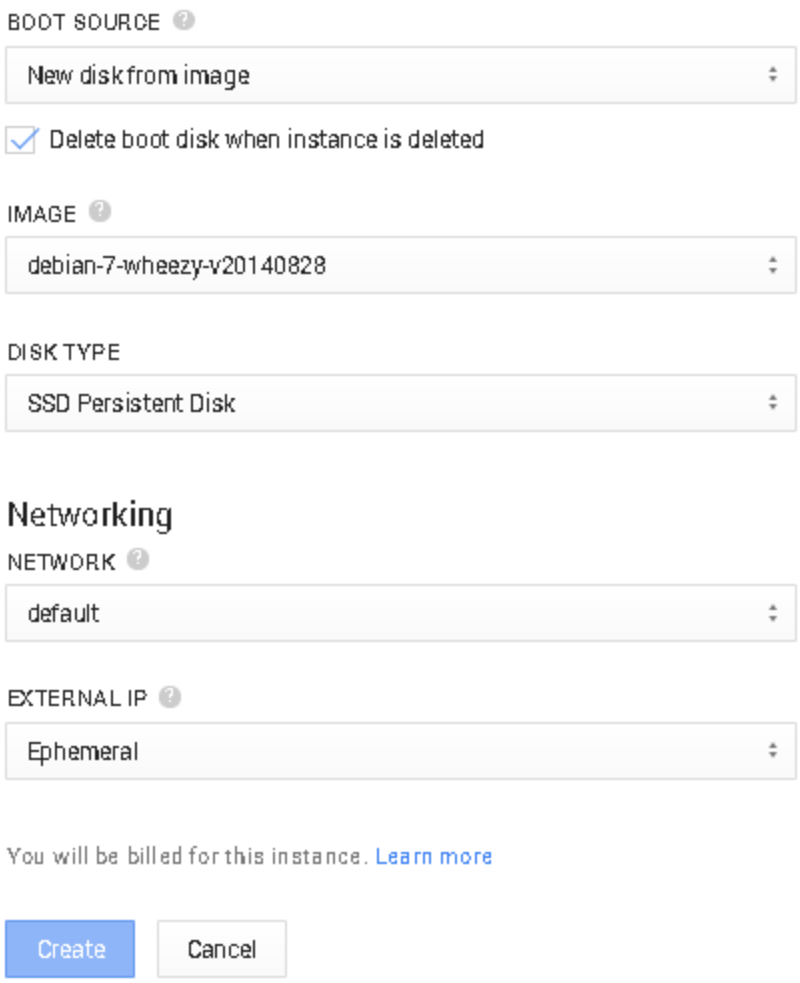
Image : It determines the Operating System installed on the system.

Network : A network performs the same function that a router does in a home network: it describes the network range and gateway IP address, handles communication between instances, and serves as a gateway between instances and callers outside the network.

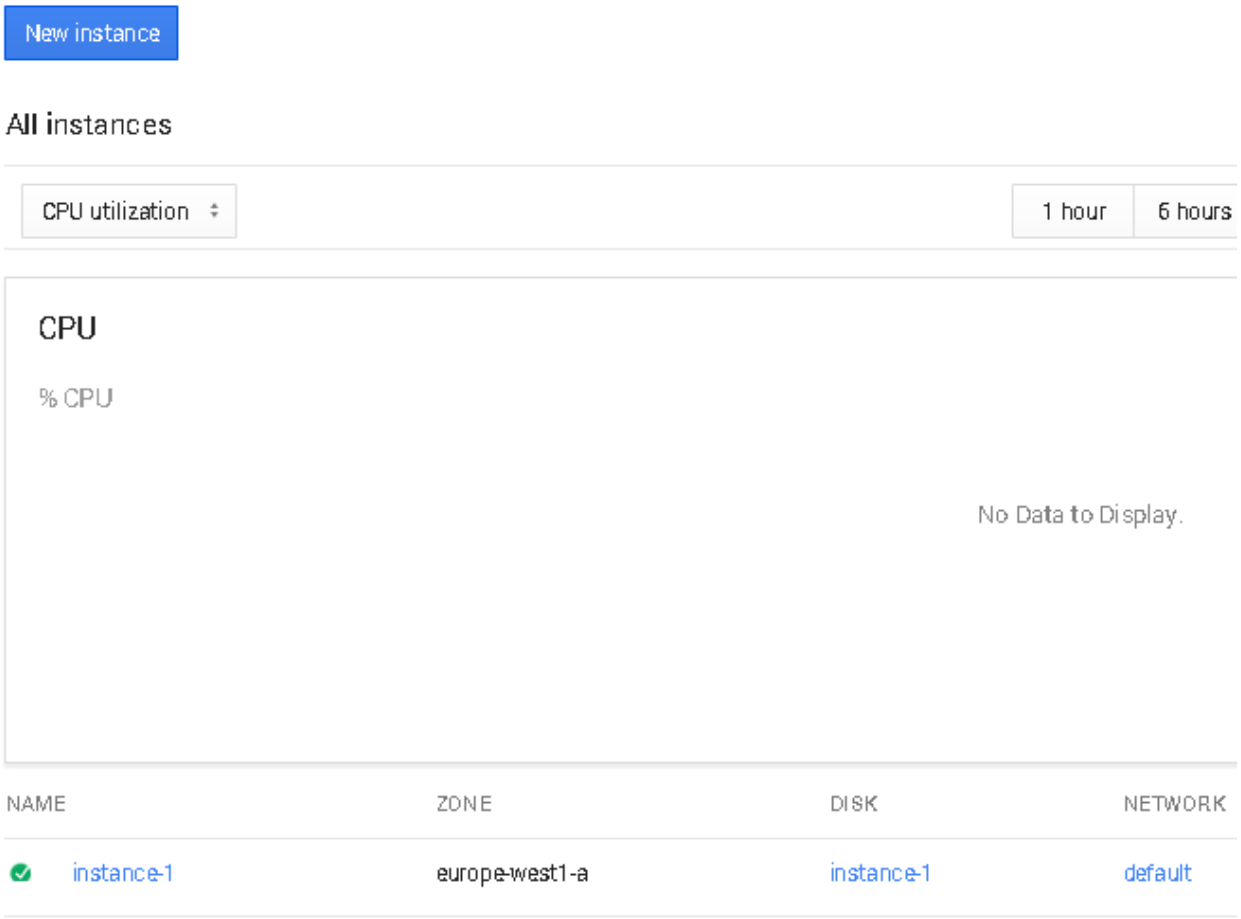
External IP : This IP Address enables the communication with outside the instance's network.

these are the values to enter to create the same machine as this video





select create, you may see some confusing screens, wait for a few minutes and you should eventually see the screen below if so you are good

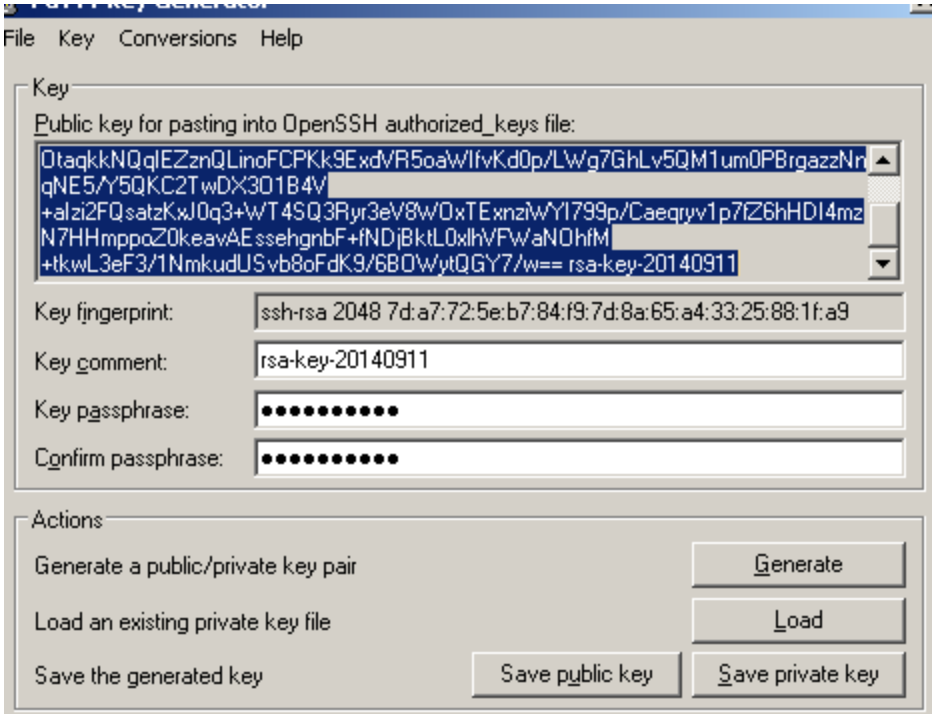


Step 4

Configure terminal access

now we will create the ssh keys to log in with putty

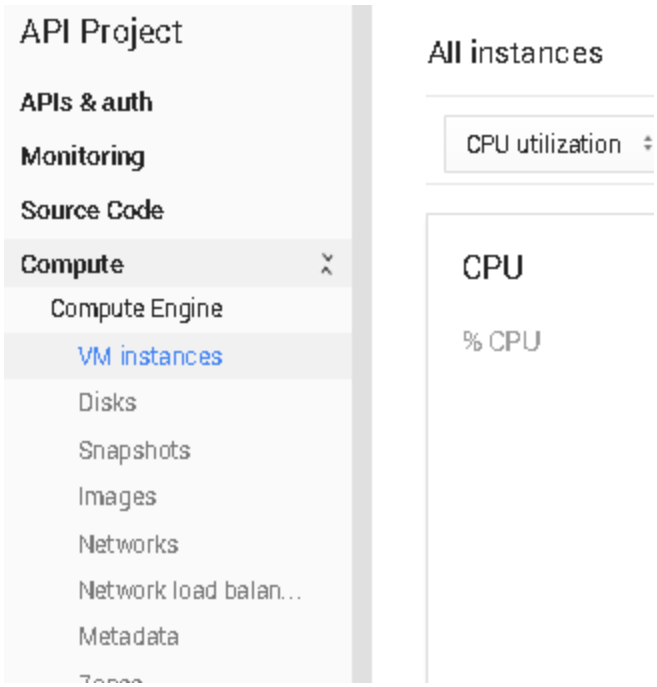
http://winscp.net/eng/docs/ui_puttygen



make sure to add a paraphrase this will be your login name with putty later

press the Save private key button

now in the developer console



select the metadata link shown above

then select the ssh keys tab

from the puttygen key display copy the raw key just as i have pasted below

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAytx42qjmJA1OJUmOZeOjw62uGER7MADYxxEcVqBclcVkZHdRfK4FDVKVInKIGlcRA0z4hBJlU5Z2
Ruw5hQhWCsCQ14vVnFbGiecREOtaqkkNQqIEZznQLinoFCPKk9ExdVR5oaWIfvKd0p/LWg7GhLv5QM1um0PBrgazzNnqNE5/Y5QKC2Tw
DX3O1B4V+aIzi2FQsatzKxJ0q3+WT4SQ3Ryr3eV8WOxTExnziWYI799p/Caeqryv1p7fZ6hHDI4mzN7HHmppoZ0keavAEssehgnbF+fN
DjBktL0xlhVFWaNOhfM+tkwL3eF3/1NmkudUSvb8oFdK9/6BOWytQGY7/w== rsa-key-20140911
```

then paste just what you have copied from puttygen like above

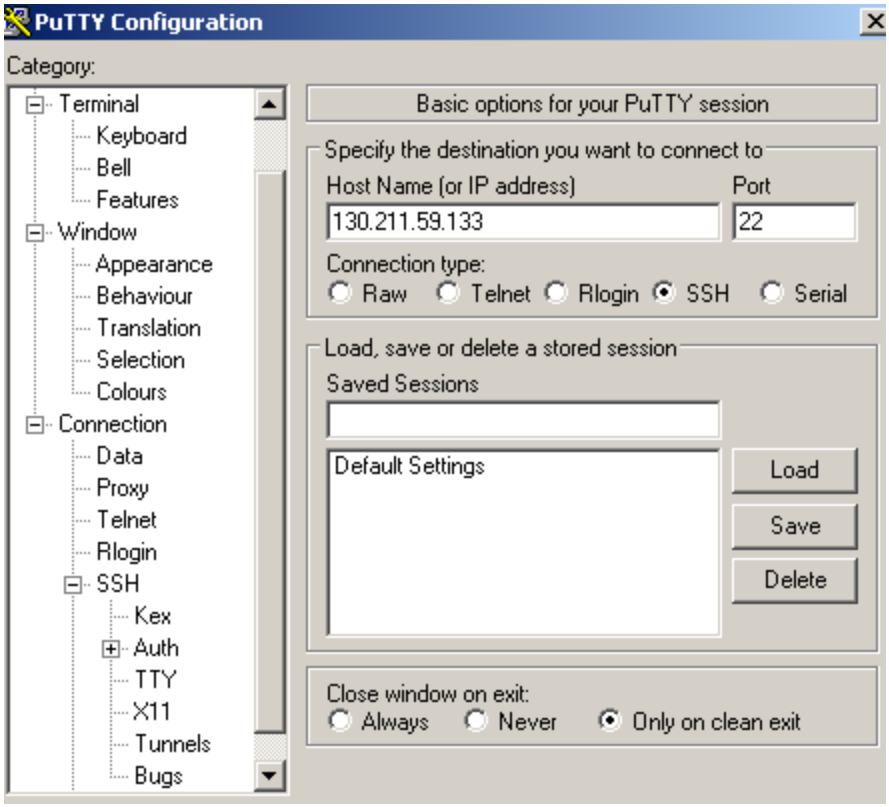
rsa-key-20140911	ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAQEAytx4… vb8oFdK9/6BOWytQGY7/w== rsa-key-20140911

rsa-key-20140911 this will be the username when we login in with putty

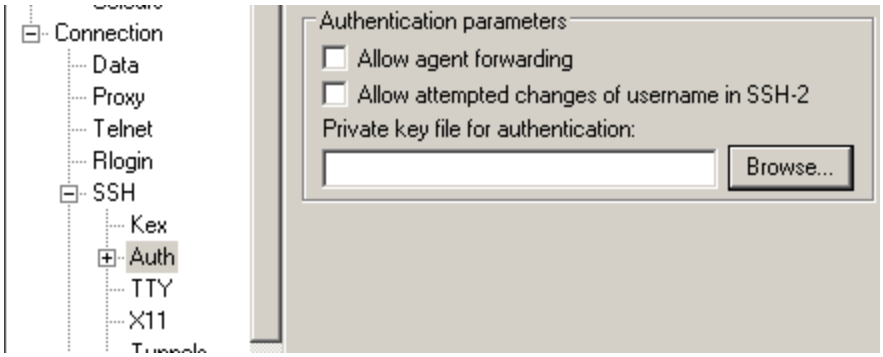
in the instance view

NAME	ZONE	DISK	NETWORK	EXTERNAL IP	CONNECT
instance-1	europe-west1-a	instance-1	default	130.211.59.133	SSH ⋮

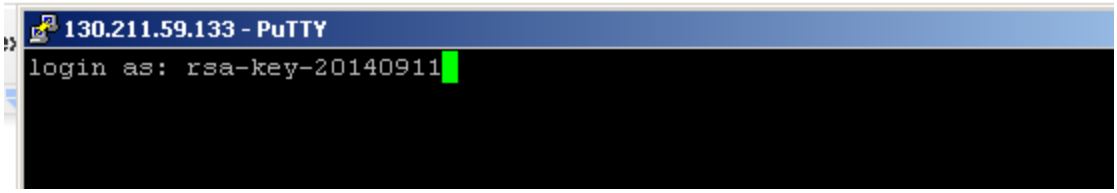
the value for the external ip 130.211.59.133 will be used as the host name in putty



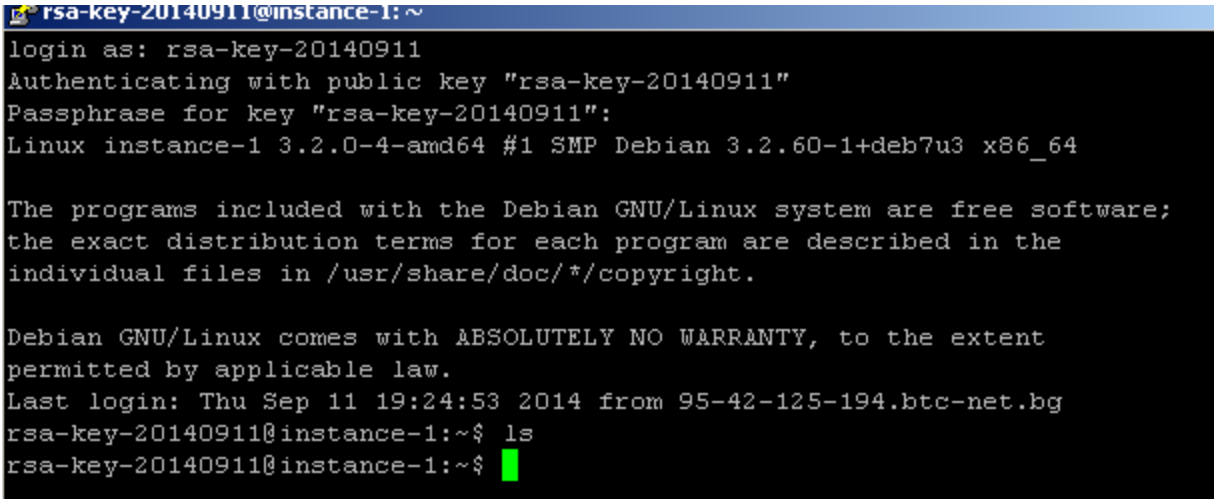
select Auth and load the key you created



return to the session view select open and use the key username as login



enter the key parahase when prompted



Step 5

Install an Application

now we have a debian box

lets install mongodb

There is a bug in the GCE Debian images where the default locale isn't set. This prevents MongoDB from starting properly from the Debian packages. The workaround is to set a default:

```
sudo locale-gen en_US.UTF-8
sudo locale-gen en_IE.UTF-8
sudo dpkg-reconfigure locales

select all i the first pop up screen and in the second screen accept the default
```

```
rsa-key-20140911@instance-1:~$ sudo locale-gen en_IE.UTF-8
Generating locales (this might take a while)...
  en_US.UTF-8... done
Generation complete.
rsa-key-20140911@instance-1:~$ sudo dpkg-reconfigure locales
Generating locales (this might take a while)...
  aa_DJ.UTF-8... done
  aa_DJ.ISO-8859-1... done
  aa_ER.UTF-8... done
  aa_ER.UTF-8@saaho... done
  aa_ET.UTF-8... done
  af_ZA.UTF-8... done
  af_ZA.ISO-8859-1... done
  am_ET.UTF-8... done
  an_ES.UTF-8... done
  an_ES.ISO-8859-15... done
  ar_AE.UTF-8... done
  ar_AE.ISO-8859-6... done
  ar_BH.UTF-8... done
  ar_BH.ISO-8859-6... done
  ar_DZ.UTF-8... done
  ar_DZ.ISO-8859-6... done
  ar_EG.UTF-8... done
  ar_EG.ISO-8859-6...
```

it will take a while

sudo apt-get update

```
sudo echo 'deb http://downloads-distro.mongodb.org/repo/debian-sysvinit dist 10gen' | tee /etc/apt/sources.list.d/mongodb.list
```

```
sudo apt-get update
```

```
sudo apt-get install -y mongodb
```

```
/etc/init.d/mongodb start
```

```
rsa-key-20140911@instance-1:~$ sudo /etc/init.d/mongodb start
[ ok ] Starting database: mongodb apparently already running.
rsa-key-20140911@instance-1:~$ mongo
MongoDB shell version: 2.0.6
connecting to: test
>
```

http://walkintocloud.com/2014/01/launching-your-first-vm-instance-on-gce/